



C2184
Úvod do programování
v Pythonu

Lekce 2

Začínáme programovat I.

Základní konstrukce jazyka: základní datové typy, matematické operátory.

*C2184 Úvod do programování v Pythonu
podzim 2015*

Proměnné

Základní datové typy

Operátory

Mgr. Stanislav Geidl
Národní centrum pro výzkum biomolekul
Masarykova univerzita

Proměnná (variable)

- má svůj název (identifikátor, identifier) `a`
- odkazuje na místo v paměti, kde je uložena tzv. hodnota (value) proměnné
- propojení proměnné a její hodnoty provádíme pomocí přiřazení (assign) (=)

`a = 1` ukázka

- Python je jazyk s dynamickou typovou kontrolou (do proměnné můžete uložit cokoliv, ale hodnoty si pamatují svůj typ)

`10 + "a"` ukázka

- název proměnné se může skládat z malých (a velkých) písmen (bez diakritiky), čísel a podtržítka (`_`), nesmí začínat číslem, podtržítkem začínají speciální proměnné, název proměnné se nesmí shodovat s žádným z 34 klíčových slov jazyka Python: `and`, `as`, `assert`, `break`, `class`, `continue`, `def`, `del`, `elif`, `else`, `except`, `finally`, `for`, `from`, `global`, `if`, `import`, `in`, `is`, `lambda`, `nonlocal`, `not`, `or`, `pass`, `raise`, `return`, `try`, `while`, `with`, `yield`, `True`, `False`, `None`



- 1 **boolean** (booleovský typ) nabývá buď hodnoty `True` nebo `False`.
- 2 Čísla mohou být celá (**integer**; 1 a 2), reálná (**float**; 1.1 a 1.2) nebo komplexní (**complex**; $3+1j$).
- 3 Řetězce (**string**) jsou posloupnosti Unicode znaků. Tuto podobu může mít například html dokument.
- 4 Bajty (**byte**) a pole bajtů, například soubor s obrázkem ve formátu jpeg.
- 5 Seznamy (**list**) jsou uspořádané posloupnosti hodnot.
- 6 N-tice (**tuple**) jsou uspořádané, neměnné posloupnosti hodnot.
- 7 Množiny (**set**) jsou neuspořádané kolekce hodnot.
- 8 Slovníky (**dictionary**) jsou neuspořádané kolekce dvojic klíč-hodnota.





- funkce je část programu (je podprogram, angl. subroutine), která lze volat opakovaně z různých částí programu, příkladem může být funkce `print()`

- funkce type()

```
>>> type(10)
<type 'int'>
>>> type(1.5)
<type 'float'>
>>> type(3+1j)
<type 'complex'>
>>> type(True)
<type 'bool'>
>>> type("test")
<type 'str'>
>>> type(None)
<type 'NoneType'>
```

- změna pomocí funkce, která nese název podle typu, který bude výsledkem, např. int()



Zjištění a změna typu

- funkce type()
- změna pomocí funkce, která nese název podle typu, který bude výsledkem, např. int()

```
>>> int("10")
10
>>> int("10a")
ValueError: invalid literal for int()
with base 10: '10a'
>>> int(10.6)
10
>>> float(1)
1.0
>>> bool(0)
False
>>> bool(1)
True
>>> bool(None)
False
>>> str(1)
'1'
```





- + sčítání (např. $10 + 3$)
- odčítání (např. $10 - 3$)
- * násobení (např. $10 * 3$)
- / dělení (např. $10/3$)
- // celočíselné dělení (např. $10 // 3 = ??$)
- % modulo (zbytek po celočíselném dělení; např. $10 \% 3 = ??$)
- ** umocňování (např. $10 ** 3$)
- () závorky
- abs () funkce, která vrací absolutní hodnotu

Proměnné

Základní datové typy

Operátory

- pořadí operací jako v matematice
- Co bude výsledkem $2 ** 8 - (5 + 5) * 5 * 5 + 5$?

Speciální typy přiřazení: $+=$ $-=$ $*=$ $/=$ $\%=$ $//=$

```
a += 1 # a = a + 1
```

Příklad - řešení kvadratické rovnice



Příklad - řešení kvadratické rovnice



$$ax^2 + bx + c = 0 \quad (1)$$

$$D = b^2 - 4ac \quad (2)$$

```
D = b**2 - 4*a*c
```

$$x_{1,2} = \frac{-b \pm \sqrt{D}}{2a} \quad (3)$$

```
x1 = (-b+math.sqrt(D)) / (2*a)
```

```
x2 = (-b-math.sqrt(D)) / (2*a)
```

- b (jedno malé písmeno)
- B (jedno velké písmeno)
- lowercase
- lower_case_with_underscores
- UPPERCASE
- UPPER_CASE_WITH_UNDERSCORES
- CapitalizedWords (nebo CapWords nebo CamelCase nebo StudlyCaps)
- mixedCase
- Capitalized_Words_With_Underscores (hnusné!)

Pro názvy lokálních proměnných budeme používat styl lowercase a lower_case_with_underscores. Speciální případ bude, když nám název proměnné bude kolidoval s klíčovým slovem např. `class_`.



Code style - přiřazení

Správně:

```
x = 1
y = 2
long_variable = 3
```

Špatně:

```
x=1
y = 2
long_variable = 3
```



Code style - volání funkce

Správně:

```
# Volání funkce bez mezery
spam(1)
# Vertikální zarovnání na otvírací závorkou
foo = long_function_name(var_one, var_two,
                          var_three, var_four)
# Parametry na dalším řádkách s vlastním
# odsazením
foo = long_function_name(
    var_one, var_two,
    var_three, var_four)
```

Špatně:

```
# Volání funkce s mezery
spam (1)
# Zákaz používat argumenty na prvním řádku,
# pokud nepoužijeme vertikální zarovnání
foo = long_function_name(var_one, var_two,
    var_three, var_four)
```





Správně:

```
i = i + 1
submitted += 1
x = x*2 - 1
hypot2 = x*x + y*y
c = (a+b) * (a-b)
```

Špatně:

```
i=i+1
submitted +=1
x = x * 2 - 1
hypot2 = x * x + y * y
c = (a + b) * (a - b)
```

Proměnné

Základní datové typy

Operátory